

Python Dependency Management: UV vs pip and venv

Managing Python dependencies is crucial for keeping your projects isolated and reproducible. This tutorial covers traditional tools (`pip` and `venv`) and introduces `uv`, a modern alternative.

1. Traditional Tools: pip + venv

What is venv?

`venv` is a built-in Python module that creates isolated environments for dependencies.

How to Use venv and pip

1. **Create a virtual environment**

```
```bash
python -m venv env
```
```

2. **Activate the environment**

- On Windows:

```
```bash
env\Scripts\activate
```
```

- On macOS/Linux:

```
```bash
source env/bin/activate
```
```

3. **Install packages with pip**

```
```bash
pip install requests
```
```

4. **Freeze dependencies**

```
```bash
pip freeze > requirements.txt
```
```

5. **Install from requirements**

```
```bash
pip install -r requirements.txt
```
```

2. Modern Tool: UV

`uv` is a fast Python package manager and virtual environment tool from the developers of `pipx` and `rye`.

Advantages of `uv`

- Much faster than pip
- Replaces both pip and venv
- More deterministic and cache-aware
- Can install from `requirements.txt` or `pyproject.toml`

Installing `uv`

```
```bash
curl -Ls https://astral.sh/uv/install.sh | sh
```
```

Or with pipx:

```
```bash
pipx install uv
```
```

Using `uv`

1. ****Install packages into a virtual environment****

```
```bash
uv venv
uv pip install requests
```
```

2. ****Run Python in the environment****

```
```bash
uv python
```
```

3. ****Freeze dependencies****

```
```bash
uv pip freeze > requirements.txt
```
```

4. ****Install from requirements****

```
```bash
uv pip install -r requirements.txt
```
```

With pyproject.toml

If using `pyproject.toml`, `uv` also supports:

```
```bash
uv pip install .
```
```

Summary

| Feature | pip + venv | uv | |
|----------|-----------------|------------------|--|
| ----- | ----- | ----- | |
| Speed | Moderate | Very Fast | |
| Built-in | Yes (in Python) | No (third-party) | |

| | | | |
|----------------|-------------------|----------------------------|--|
| Ease of Use | Manual activation | Simplified workflow | |
| Locking | `pip freeze` | `uv pip freeze` | |
| Modern tooling | No | Yes (supports `pyproject`) | |

Use `pip` + `venv` if you want the standard, built-in solution.

Use `uv` if you want speed and a streamlined workflow.